



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2), Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia. Meskipun komputer berasal dari kata “Komputasi”, komputasi yang memang dilaksanakannya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan komputer untuk membuat dokumen, berbagai perhitungan yang dilakukan tidak terlihat. Sebagai contoh, ketika pemakai memilih pengaturan “*justify*” atau rata di batas kiri dan batas kanan dokumen, perhitungan untuk menambahkan karakter-karakter spasi sebenarnya terjadi.

Noersasongko dan Andono (2010:2), Komputer adalah serangkaian atau sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasarkan urutan instruksi ataupun program yang diberikan kepada Komputer.

Dari definisi diatas penulis menyimpulkan bahwa komputer adalah merupakan perangkat keras yang dapat digunakan untuk membantu dalam berbagai pekerjaan manusia seperti pembuatan dokumen

2.1.2 Pengertian Perangkat Lunak

Kadir (2017:2), Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada *computer* agar dapat melaksanakan tugas sesuai kehendak pemakai. System operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi seperti *Microsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak.

Simarmata (2010:2), Perangkat lunak adalah instruksi langsung komputer untuk melakukan pekerjaan dan dapat ditemukan di setiap aspek kehidupan modern dari aplikasi yang kritis untuk hidup (*life-critical*), seperti perangkat



pemantauan medis dan pembangkit tenaga listrik, sampai perangkat hiburan, seperti video game. Banyak produk perangkat lunak berisi jutaan baris kode yang diharapkan dapat melakukan pekerjaan dengan baik dalam menghadapi perubahan kondisi. Semua perangkat lunak juga membutuhkan keandalan yang tinggi dan harus dihasilkan secara ekonomis.

Dari definisi diatas penulis menyimpulkan bahwa perangkat lunak adalah suatu program aplikasi yang digunakan sesuai dengan perintah yang ditujukan untuk menjalankan tugasnya.

2.1.3 Pengertian Perangkat Keras

Menurut Sutabri (2014:32), “Perangkat keras sebagai subsistem dari sistem komputer juga mempunyai komponen, yaitu

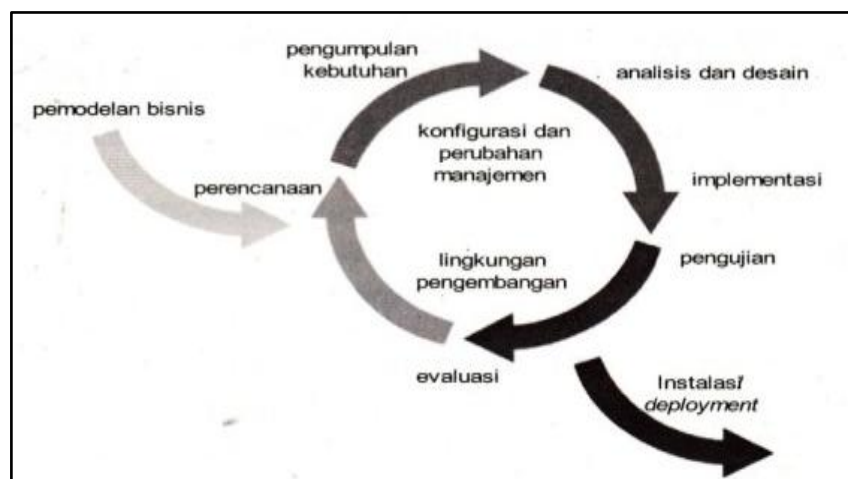
1. Komponen peranti masukan (*input device*) yaitu alat yang digunakan untuk menerima masukkan data atau masukkan program misalnya *keyboard, scanner, sensor*, dan pengenalan suara.
2. Komponen peranti pemroses (*processing device*), yaitu alat di mana instruksi-instruksi program dieksekusi untuk memproses data yang dimasukkan lewat alat masukan yang hasilnya nanti akan ditampilkan dialat output misalnya *CPU (central processing unit)*.
3. Komponen peranti keluaran (*output device*), yaitu alat untuk menampilkan pengolahan data seperti tulisan (huruf, kata, angka, karakter, dan simbol-simbol), *image* (bentuk grafik atau gambar) dan suara (bentuk musik) misalnya *hardcopy device (printer)* dan *softcopy device (monitor)*.
4. Komponen peranti penyimpanan (*storage*) eksternal adalah penyimpanan luar, karena terletak diluar alat prosesnya atau disebut *mass storage* atau penyimpanan masal karena penyimpanan umumnya lebih besar dari *main memory*”. Lain halnya menurut Kadir (2017:2), “Perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik. Monitor, *hard disk*, dan *mouse* adalah contoh perangkat keras”.



Dari beberapa definisi perangkat keras penulis menyimpulkan bahwa perangkat keras adalah beberapa komponen yang bekerja saling mendukung sesuai dengan instruksi *software*.

2.1.4 Metode Pengembangan Sistem

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), RUP (Rational Unified Process) merupakan pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar 2.1.



(Sumber: Sumber: Sukamto dan Shalahuddin dalam (Fitria dan Widiowati, 2017:28))

Gambar 2.1 . Proses Iteratif RUP

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), Fase RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut adalah penjelasan untuk setiap fase pada RUP dalam pengembangan aplikasi administrasi kependudukan.



1. *Fase Inception*

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), Pada fase ini, menyimpulkan beberapa kegiatan yang dilakukan berdasarkan hal-hal diatas, diantaranya penyelidikan awal, studi kelayakan hingga kebutuhan fungsional dan non fungsional.

1. Pengertian Penyelidikan Awal

Menurut Yahaya dkk (2006:2), Penyelidikan pada hakikatnya adalah bagi mencari jawaban atau menyelesaikan dengan jawaban yang benar atau sekurang-kurangnya dapat mengetahui kebenaran yang logik melalui pemikiran yang mempunyai kesinambungan dengan fakta empiric.

Pada akhir abad ke-19, penyelidikan banyak dilakukan dalam bidang ilmu pengetahuan alam. Penyelidikan pada ketika itu dilaksanakan dengan tujuan mengembangkan teori-teori dan konsep-konsep ilmu yang sedia ada. Penyelidikan-penyelidikan yang dilaksanakan banyak menggunakan kaedah ilmiah, yaitu melalui penerapan langkah-langkah berfikir secara ilmiah dan mendalam.

2. Studi Kelayakan

Menurut Jogiyanto dalam Widiyanto dan Syaifullah (2014:201) Suatu studi kelayakan (Feasibility study) adalah suatu studi yang akan digunakan untuk menentukan kemungkinan apakah pengembangan proyek sistem layak diteruskan atau dihentikan. Studi kelayakan disebut juga dengan istilah High point review.

3. Pengertian Kebutuhan Fungsional

Menurut Kosasi & Kuway dalam melinda dkk (2017:2), kebutuhan fungsional adalah pernyataan layanan yang harus diberikan kepada sistem agar dapat melakukan keperilakuannya dalam bereaksi terhadap masukan tertentu dan pada situasi tertentu. Kebutuhan fungsional harus dapat mengilustrasikan secara jelas fungsi-fungsi dan fitur-fitur yang ada pada sistem yang dikembangkan



4. Pengertian Kebutuhan Non Fungsional

Menurut Kosasi & Kuway, (2012) dalam melinda dkk (2017:2), kebutuhan non-fungsional berisi tentang batasan layanan atau fungsi yang diberikan sistem. Kebutuhan non-fungsional ini mencakup hardware dan software yang digunakan.

2. *Fase Elaboration*

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), Pada tahap ini dari hasil observasi dan wawancara tersebut penulis dapat melakukan identifikasi masalah pada sistem yang dibuat. Didalam elaboration terdapat dua tahapan yaitu :

1. Analisis, Terdapat tiga fase dalam tahapan analisis sistem pada alur pengembangan sistem RUP, yaitu: analisis permasalahan, analisis persyaratan, dan analisis keputusan.
2. Perancangan, Pada tahap perancangan terdiri dari: perancangan aplikasi, menggunakan diagram UML meliputi use case diagram, perancangan tampilan, dan menggunakan struktur navigasi.

3. *Fase Construction*

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), Pada tahap ini menjelaskan bagaimana mengimplementasi dan melakukan uji coba terhadap aplikasi yang telah dibuat. Dalam tahapan implementasi dijelaskan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengimplementasi aplikasi ini. Sedangkan pada tahapan uji coba dilakukan testing. Testing diperlukan untuk menjamin kualitas aplikasi yang telah dibuat apakah telah sesuai dengan yang diharapkan.

1. Pengertian Testing

Menurut Romeo dalam Putri (2020:21), Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (defects / errors / bugs) dan mengevaluasi fitur-fitur dari entitas software. Testing software adalah proses



mengoperasikan software dalam suatu kondisi yang dikendalikan, untuk verifikasi, mendeteksi error, dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk software, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi error adalah testing yang seharusnya berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut tidak terjadi.

2. Pengertian *Black Box Testing*

Menurut Romeo dalam jurnal Putri (2020:21), *Black-box* testing dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Dengan adanya *black box testing*, perekayasa *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari error dan fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan. Kategori error yang akan diketahui melalui *black box testing*, yaitu:

- a. Fungsi yang hilang atau tak benar
- b. Error dari antar-muka
- c. Error dari struktur data atau akses eksternal database
- d. Error dari kinerja atau tingkah laku
- e. Error dari inisialisasi dan terminasi

4. *Fase Transition*

Menurut Sukamto dan Shalahuddin dalam Fitria dan Widowati (2017:28), Pada tahap *transition* membuat panduan penggunaan dari aplikasi yang telah dibuat.



2.2 Teori Khusus

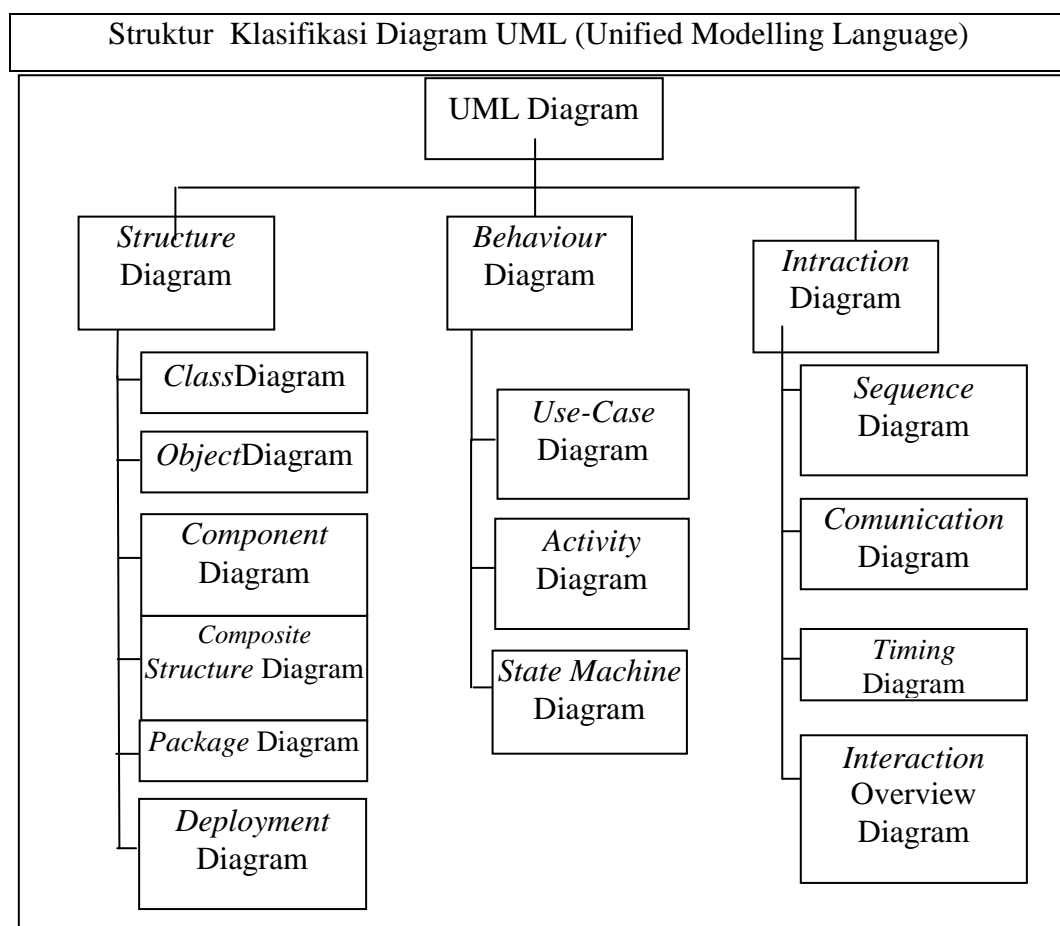
2.2.1 Pengertian UML (*Unified Modeling Language*)

Sukanto dan Shalahuddin (2016:133), UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.2.2 Macam - macam diagram *Unified Modeling Language* (UML)

Sukanto dan Shalahuddin (2016:140), Pada UML terdapat 13 buah diagram yang selanjutnya dikelompokkan menjadi 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:

Struktur Klasifikasi Diagram UML (Unified Modelling Language)



(Sumber: Sukanto dan Shalahuddin, 2016:140)

Gambar 2.2 Klasifikasi Diagram UML (*Unified Modelling Language*)



Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sebuah sistem yang dimodelkan.

b. *Behavior Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan sebuah interaksi sistem dengan sistem lain nya ataupun interaksi antar subsistem pada suatu sistem.

2.2.3 Pengertian *Class Diagram*





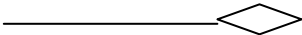
Sukanto dan Shalahuddin (2016:141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- Operasi atau model adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2.1 Simbol-simbol dalam *class diagram*

Simbol	Deskripsi
<div>Kelas</div> <div> <div>Nama_kelas</div> <div>+atribut</div> <div>+operasi()</div> </div>	Kelas pada struktur system
<div>Antarmuka/<i>interface</i></div> <div> <div>○</div> <div>Nama_<i>interface</i></div> </div>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Lanjutan Tabel 2.1 Simbol-simbol dalam *class diagram*

Asosiasi/ <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/ <i>aggregation</i> 	Relasi antarkelass dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Sukamto dan Shalahuddin, 2016:146-147)

2.2.4 Pengertian *Use Case Diagram*

Sukamto dan Shalahuddin (2016:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

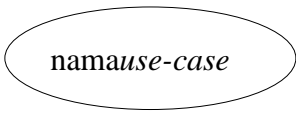
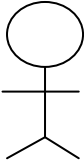
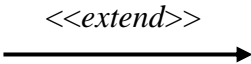
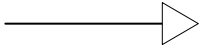
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *usecase*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.



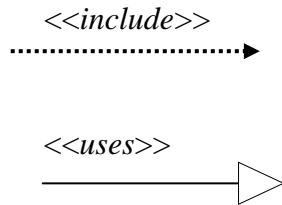
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.2. Simbol-simbol *Use-Case* Diagram

No	Simbol	Deskripsi
1.	<p><i>Use-Case</i></p>  <p>namause-case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use-case</i> .
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor nama_interface</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
4.	<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use-case</i> tambahan ke sebuah <i>use-case</i> dimana <i>use-case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use-case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>use-case</i> yang ditambahkan; biasanya <i>use-case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use-case</i> yang menjadi induknya.
5.	<p>Generalisasi/<i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use-case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

**Lanjutan Tabel 2.2.** Simbol-simbol *Use-Case* Diagram

6.	Menggunakan / <i>include</i> / <i>uses</i>	Relasi <i>use-case</i> tambahan ke sebuah <i>use-case</i> di mana <i>use-case</i> yang ditambahkan memerlukan <i>use-case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use-case</i> .
----	--	--



(Sumber: Sukamto dan Shalahuddin (2016:156-158))


2.2.5 Pengertian *Activity* Diagram

Sukamto dan Shalahuddin (2016:161), *Activity* Diagram atau Diagram Aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:





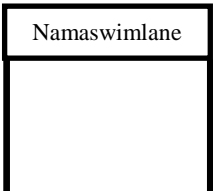
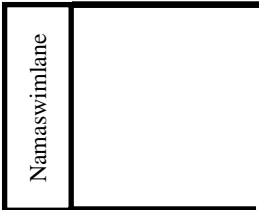
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak

Tabel 2.3 Simbol-simbol dalam *Activity* Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal



Lanjutan Tabel 2.3 Simbol-Simbol Dalam Activity Diagram

Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber: Sukamto dan Shalahuddin, 2016:162-163)




2.2.6 Pengertian *Sequence Diagram*



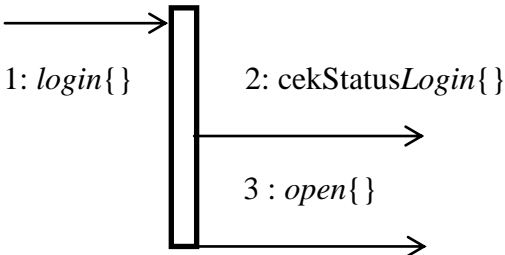

Sukanto dan Shalahuddin (2016:165), Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat sekenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

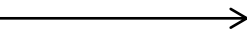
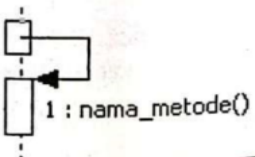
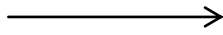
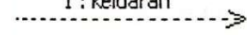
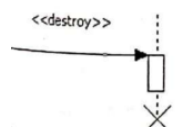
Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Namaaktor</div> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i></p>

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

Garishidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek <div style="border: 1px solid black; padding: 2px; width: fit-content;"> Nama objek :nama kelas </div>	Menyatakan objek yang berinteraksi pesan
Waktuaktif 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>Maka cek Status <i>Login{ }</i> dan <i>open{ }</i> dilakukan di dalam metode <i>login{ }</i></p> <p>Aktor tidak memiliki waktu aktif</p>
Pesantipe <i>create</i> <i><<create>></i> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada arah objek yang dibuat
Pesantipe <i>call</i> 1 :nama_metode{ }	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

	 <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesantipe <i>send</i></p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesantipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

(Sumber: Sukamto dan Shalahuddin, 2016:165-167)

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Koyuko dkk dikutip Oki (2016), Aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen,



penambahan data dan beberapa tugas lainnya. *Suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. Salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut dan juga aplikasi-aplikasi tersebut mempunyai bisa berinteraksi antar aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.

Menurut Juansyah (2015:2), Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer* eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Dari definisi diatas penulis menyimpulkan bahwa aplikasi adalah suatu software yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya, serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.

2.3.2 Pengertian Penerimaan

Menurut Kamus Besar Bahasa Indonesia dalam situsnya <https://kbbi.kemdikbud.go.id/entri/penerimaan>, Penerimaan adalah proses, cara, perbuatan menerima; penyambutan.

Menurut Desi Anwar dalam Solihin dan Witanto (2016:55), Penerimaan merupakan penyambutan, proses, perbuatan atau sikap terhadap seseorang. Siswa merupakan pelajar pada akademi atau perguruan tinggi. Baru merupakan suatu hal belum ada sebelumnya.



Dari definisi diatas penulis menyimpulkan bahwa penerimaan adalah proses penyambutan kepada pelajar dalam memulainya proses kegiatan yang belum ada sebelumnya.

2.3.3 Pengertian Mahasiswa

Menurut Kamus Besar Bahasa Indonesia dalam situsnya <https://kbbi.web.id/mahasiswa>, Mahasiswa adalah orang yang belajar di perguruan tinggi.

Menurut Sarwono dalam Anwar (2017:26), Mahasiswa adalah setiap orang yang secara terdaftar untuk mengikuti pelajaran disebuah perguruan tinggi dengan batasan umur sekitar 18 – 30 tahun. Mahasiswa merupakan suatu kelompok dalam masyarakat yang memperoleh statusnya, karena adanya ikatan dengan suatu perguruan tinggi

Dari definisi diatas penulis menyimpulkan bahwa Mahasiswa adalah dapat didefinisikan sebagai individu yang sedang menuntut ilmu ditingkat perguruan tinggi, baik negeri maupun swasta atau lembaga lain yang setingkat dengan perguruan tinggi.

2.3.4 Pengertian PKL (Praktek Kerja Lapangan)

Dalam Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia Nomor Per.22/MEN/IX/2009 Tentang Penyelenggaraan Pemagangan di Dalam Negeri. Pemagangan adalah bagian dari sistem pelatihan kerja yang diselenggarakan secara terpadu antara pelatihan di lembaga pelatihan dengan bekerja secara langsung di bawah bimbingan dan pengawasan instruktur atau pekerja yang lebih berpengalaman dalam proses produksi barang atau jasa di perusahaan, dalam rangka menguasai keterampilan atau keahlian tertentu.

Menurut Wikipedia dalam situsnya https://id.wikipedia.org/wiki/Pelatihan_kerja_lapangan, Pelatihan kerja lapangan, Praktik kerja lapangan, on-the-job training atau biasa disebut dengan PKL adalah salah satu bentuk kegiatan yang bertempat di lingkungan kerja langsung. PKL bisa dilakukan oleh murid SMA/SMK, mahasiswa maupun karyawan baru. Di tingkat



mahasiswa, PKL merupakan implementasi secara sistematis dan sinkron antara program pendidikan di sekolah dengan program penguasaan keahlian yang diperoleh melalui kegiatan kerja secara langsung di dunia kerja untuk mencapai tingkat keahlian tertentu.

Dari definisi diatas penulis menyimpulkan bahwa PKL adalah adalah salah satu bentuk kegiatan yang bertempat di lingkungan kerja langsung, yang diselenggarakan secara terpadu antara pelatihan di lembaga pelatihan dengan bekerja secara langsung di bawah bimbingan dan pengawasan instruktur atau pekerja yang lebih berpengalaman dalam proses produksi barang atau jasa di perusahaan, dalam rangka menguasai keterampilan atau keahlian tertentu.

2.3.5 Pengertian Web

Abdulloh (2018:1), *Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standard yaitu HTML. Skrip HTML ini akanditerjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

Hidayat dalam jurnal Nurmi (2017:2), Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing- masing dihubungkan dengan jaring-jaringan halaman.

Dari definisi diatas penulis menyimpulkan bahwa website dibuat menggunakan bahasa standard yaitu HTML. Skrip HTML ini akan diterjemahkan oleh web browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang seperti teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang



membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaring-jaringan halaman.

2.3.6 Pengertian Aplikasi Penerimaan Mahasiswa PKL (Praktek Kerja Lapangan) Pada PT. Bukit Asam Tbk Unit Dermaga Kertapati Palembang Berbasis Web

Dimana sistem ini digunakan untuk melakukan penerimaan mahasiswa PKL dalam hal ini secara tidak langsung aplikasi juga dapat mendaftarkan mahasiswa yang pernah melakukan PKL di PTBA Unit Dermaga Kertapati. Pada penelitian ini akan dibangun aplikasi yang dapat digunakan untuk memudahkan mahasiswa dalam mengajukan PKL dan menerima pemberitahuan tentang diterima atau tidaknya pengajuan PKL yang dilakukan. Aplikasi yang dibuat haruslah dapat diakses melalui web sehingga dapat diakses secara langsung oleh mahasiswa ketika terhubung dengan Internet.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Fathansyah (2018:2), Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Menurut Winarno dan Utomo dalam Prayitno dan Safitri (2015:2), “Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling



berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada”.

Dari definisi diatas penulis menyimpulkan bahwa Basis Data kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah

2.4.2 Pengertian MySQL

Menurut Wahana Komputer dalam firman (2016:30), MySQL adalah database server open source yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas API (Application Programming Interface yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

Menurut Anhar (2010:21) dalam Prayitno dan Safitri (2015:2), “MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL Database Management System atau DBMS dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL dan lainnya”.

Dari definisi diatas penulis menyimpulkan bahwa *MySQL* adalah software database yang digunakan oleh praktisi untuk membangun suatu project dengan banyak DBMS seperti Oracle, MS SQL, Postagre SQL dan lainnya.

2.4.3 Pengertian Bahasa Pemrograman

Menurut Wahyudi dkk (2013:65), Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh



komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari: Wahyudi dkk (2013:65)

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Ingggris Assembly), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <>, &&, ||, dsb.
4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur katakata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb.

Menurut Suprpto dkk (2008:30), Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer). Bahasa ini dibagi menjadi tiga tingkatan yaitu bahasa mesin, bahasa tingkat rendah dan bahasa tingkat tinggi. Bahasa mesin (*machine language*) berupa *microinstruction* atau *hardwire*. Programnya sangat panjang dan sulit dipahami. Di samping itu sangat tergantung pada arsitektur mesin. Keunggulannya adalah prosesnya sangat cepat dan tidak perlu interpreter atau penterjemah Bahasa tingkat rendah (*low level language*) berupa *macroinstruction* (*assembly*). Seperti halnya bahasa mesin, bahasa tingkat rendah tergantung pada arsitektur mesin. Programnya panjang dan sulit dipahami walaupun prosesnya cepat. Jenis bahasa tingkat ini perlu penterjemah berupa assembler.

Bahasa tingkat tinggi (*high level language*) menyerupai struktur bahasa manusia sehingga mudah dipahami. Bahasa ini tidak tergantung pada arsitektur mesin tetapi memerlukan penterjemah berupa compiler atau interpreter. Secara



garis besar ada dua kategori bahasa pemrograman yaitu: bahasa pemrograman aras rendah (*low level*) dan bahasa pemrograman level tinggi (*high level*).

Bahasa pemrograman aras rendah cenderung mendekati level komputer, ini artinya bahwa bahasanya ditulis mendekati atau sama dengan bahasa mesin komputer, hal ini sangat sulit ditulis karena bahasanya jauh dari bahasa manusia yang digunakan sehari-hari. Bahasa pemrograman yang lebih mudah dipelajari adalah bahasa pemrograman aras tinggi. Disebut aras tinggi karena bahasanya mendekati level bahasa manusia sehingga lebih mudah dipahami.

Dari definisi diatas penulis menyimpulkan bahwa Bahasa Pemrograman Bahasa ini dibagi menjadi tiga tingkatan yaitu bahasa mesin, bahasa tingkat rendah dan bahasa tingkat tinggi. Bahasa mesin berupa *microinstruction* atau *hardwire*. Programnya sangat panjang dan sulit dipahami. Di samping itu sangat tergantung pada arsitektur mesin. Keunggulannya adalah prosesnya sangat cepat dan tidak perlu interpreter atau penterjemah Bahasa tingkat rendah berupa *macroinstruction*.

2.4.4 Pengertian HTML

Abdulloh (2018:7), HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaanya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*. HTML berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai layout yang diinginkan.

HTML biasanya disimpan dalam sebuah file berekstensi *.html*. untuk mengetikkan skrip HTML dapat menggunakan *texteditor* seperti Notepad sebagai bentuk paling sederhana atau text editor khusus yang dapat mengenali setiap unsur skrip HTML dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++, *Sublime Text*, dan masih banyak lagi aplikasi lain yang sejenis.

Menurut Winarno dan Utomo dalam Prayitno dan Safitri (2015:2), “HTML singkatan dari *Hypertext Markup Language* dan berguna untuk menampilkan halaman web”.



Dari definisi diatas penulis menyimpulkan bahwa HTML berupa tag-tag yang menyusun setiap elemen dari *website*. HTML menggunakan *texteditor* seperti Notepad sebagai bentuk paling sederhana atau text editor khusus yang dapat mengenali setiap unsur skrip HTML dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++. *Sublime Text*.

2.4.5 Pengertian PHP

Menurut Binarso dkk (2012:76), “PHP (*Hypertext Preprocessor*) merupakan bahasa scripting yang tergabung menjadi satu dengan HTML dan dijalankan pada *server side* atau semua perintah yang diberikan akan secara penuh dijalankan pada *server*, sedangkan yang dikirim ke klien (*browser*) hanya berupa hasilnya saja.

Sintak dan semantic PHP memiliki kesamaan dengan bahasa C, JAVA, PERL, dengan menambahkan tambahan *tag* `<?php` sebagai pembuka dan diakhiri dengan `?>`, pasangan kedua kode inilah yang berfungsi sebagai *tag* kode PHP. Berdasarkan *tag* inilah pihak *server* dapat memahami kode PHP dan kemudian memprosesnya, hasilnya dikirim ke *browser*.

Menurut Anhar dalam Prayitno dan Safitri (2015:2), “PHP adalah (PHP Hypertext Preprocessor) adalah bahasa pemrograman web berupa script yang dapat diintegrasikan dengan HTML”.

Dari definisi diatas penulis menyimpulkan bahwa PHP adalah bahasa pemrograman web yang tergabung menjadi satu dengan HTML dan dijalankan pada *server side* atau semua perintah yang diberikan akan secara penuh dijalankan pada *server*.

2.4.6 Pengertian Web Server

Menurut Fathansyah dalam Prayitno dan Safitri (2015:2), Menerangkan bahwa pengertian web server adalah “Server Web (Web Server) merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.



Menurut Josi (2017:52), Web Server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan web *browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

Dari definisi diatas penulis menyimpulkan bahwa Web Server adalah perangkat lunak yang menyediakan layanan akses klien yang dikenal dengan web *browser*/halaman web.

2.4.7 Pengertian Xampp

Menurut Wahana dalam Prayitno dan Safitri (2015:2), “XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”.

Menurut Heriyanto dalam Josi (2017:52), Xampp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan Xampp ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat website secara offline untuk masa coba-coba di komputer sendiri. Jadi fungsi dari Xampp server itu sendiri merupakan server website kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakses web, untuk itu komputer kita harus menjadi server

Dari definisi diatas penulis menyimpulkan bahwa Xampp adalah aplikasi tools untuk menyediakan paket lunak yang berisi konfigurasi Web Server, Apache, PHP, MySQL untuk membantu kita dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web.

2.4.8 Pengertian phpMyAdmin

Nugroho (2014:10), phpMyAdmin adalah aplikasi manajemen *databaseserver* MySQL berbasis *web*. Dengan aplikasi phpMyAdmin kita bisa mengelola *database* sebagai **root**(pemilik server) juga disebut *Administrator database*. Lewat fasilitas phpMyAdmin, Anda bias membuat *database* baru, *table* dan struktur data mengelola data di dalam *database*.



Menurut Wikipedia dalam situsnya <https://id.wikipedia.org/wiki/PhpMyAdmin>, phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui website Jejaring Jagat Jembar (World Wide Web) . phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perizinan (permissions), dan lain-lain).

Dari definisi diatas penulis menyimpulkan bahwa phpMyAdmin adalah aplikasi manajemen database server MySQL berbasis web. Dengan aplikasi phpMyAdmin kita bias mengelola database sebagai root juga disebut Administrator database. Lewat fasilitas phpMyAdmin, Anda bisa membuat database baru, table dan struktur data mengelola data di dalam database.

2.4.9 Pengertian Javascript

Menurut Binarso dkk (2012:76), Javascript adalah bahasa yang berbentuk kumpulan skrip yang fungsinya digunakan untuk menambahkan interaksi antara halaman *web* dengan pengunjung halaman *web* .Javascript dijalankan pada sisi klien yang akan memberikan kemampuan fitur-fitur tambahan halaman *web* yang lebih baik dibandingkan fitur-fitur yang terdapat pada HTML.

Menurut Sidik dalam Prayitno dan Safitri (2015:2), menjelaskan bahwa “JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif, tidak sekedar indah saja”.

Dari definisi diatas penulis menyimpulkan bahwa Javascript adalah bahasa pemrograman berbentuk kumpulan skrip yang fungsinya digunakan untuk menambahkan interaksi antara halaman *web* dengan pengunjung halaman *web*.